ITOS CFDP User's Guide

Integrated Test & Operations System
5 October 2006

Table of Contents

Introduction

This document describes the ITOS CCSDS File Delivery Protocol (CFDP) application `itos_cfdp` which handles bi-directional file transfers. The application is based on the reference CFDP implementation library written by Tim Ray of Goddard Space Flight Center, Code 584.

Socket connections are configured with command line options using the ITOS socket library. The CFDP PDUs may be delivered to and accepted from the CFDP application via the ITOS telemetry and command subsystems, or directly to the ITOS CFDP application.

Multiple simultaneous bidirectional file transfers are permissible.

Chapter 1, CFDP Interfaces discusses inputs and outputs. This information is needed for configuring those interfaces at startup time. Chapter 2, CFDP Invocation discusses starting and configuring the CFDP application via command line options. Chapter 3, CFDP Directives, discusses run-time control. Chapter 4, ITOS Issues, discusses ITOS items that are needed to run the CFDP application.

This is written for Build 1 (March 2006). Additional functionality is planned.

1 CFDP Interfaces

The application maintains several socket connections. The PDU input connections may be connected to the frame_sorter over a TCP socket or may be configured directly to the external PDU data source (using either a TCP or UDP socket). It may transmit CCSDS telecommand packets to the fopmux over a TCP socket or may send the telecommand packets directly to an external recepient using either a TCP or UDP socket. If the commands go out via the fopmux the fopmux responses are read back. It receives ASCII requests (and transmits ASCII responses) to STOL over a TCP socket. It outputs event messages using the ITOS events API. The ITOS database provides command formats needed for encapsulation of the outgoing PDUs and maintains status variables (as telemetry mnemonics) in the database.

1.1 PDUs In

The input socket(s) are configurable as command line options (see Chapter 2 on Invocation for further information). Options include TCP/UDP, port number, host (if TCP client or UDP), server/client (for TCP only) retry frequency and timeout. Ultimately, up to four input socket connections will be supported. For Build 1 (March 2006) only a single socket connection has been tested. Since these connection(s) are configurable the connection may be to either the ITOS frame_sorter or to an extrnal source. At this writing only CCSDS telemetry packets or raw PDUs are supported. Input of AOS (CCCSDS Version 2) telemetry frames will be implemented in the next build. The currently configured version does not use the ITOS frame_sorter due to performance concerns with having two ITOS applications that require simultaneous high rate disk I/O. This decision will be revisited once final hardware selections are made.

The CFDP application can initiate or listen for this connection from the frame_sorter or any other CFDP data source. If the CFDP data is to be routed thourgh the frame_sorter then the STOL procedure or other script which starts the application arranges for the frame_sorter end of the connection, and tells the frame_sorter what AppIDs to send on the connection.

If the connection breaks, the application should try to re-establish it as directed by command line arguments.

1.2 PDUs Out

As with PDU input, PDU output is fully configurable as to TCP/UDP, port number, host (if TCP client or UDP), server/client (for TCP only), retry frequency and timeout. PDUs may be transmitted via the ITOS command subsystem or directly to an external receiver. If the PDUs are sent to the ITOS command subsystem the fop mux is the recipient. The transmitter in the command subsysem is cmd_transmit. cmd_transmit sends telecommand packets or telecommand frames. That selection is controlled by the setting of the ITOS_TRANSMIT_HDR variable in the itosrc file. A value of 'notf' means "no transfer frame" so packets are sent. For Build 1 (March 2006) only packets or raw PDUs are supported.

The CFDP application initiates a TCP connection to the ITOS command system, which is always listening if it is enabled. Alternatively, if bypassing the command subsytem a connection is made as configured. The bypass mode is for testing purposes only. The operational configuration routes all outgoing PDUs (in CCSDS telecommand packets) through the command subsystem.

If the connection breaks, the application retries as directed by command line arguments.

1.3 Requests In / Responses Out

The CFDP application can initiate or listen for a connection with STOL. The STOL procedure or other script which starts the application arranges for STOL's end of the connection.

The STOL `open`, `close`, `read`, and `write` directives are used from STOL to establish the connection, send directives and read responses.

The cfdp_ground.proc STOL proc provided opens a STOL connection to *cfdp_app* once the application is up. All STOL directives should reference that global variable to communicate with the CFDP process.

1.4 Events Out

The CFDP application uses the ITOS event subsystem to send messages to the user. The messages appear in the ITOS event window.

1.5 Status Out

The CFDP application provides the following status to the user:

- number of active files being transmitted
- number of active files being received
- number of files frozen
- number of files suspended
- number packets received
- number bad packets received
- number idle packets received
- number packets transmitted
- number packets failed to be transmitted
- number packets received
- elapsed time
- number bytes transmitted
- number bytes received

2 CFDP Invocation

Command line options are used in a STOL proc or other script to start the CFDP application. The next chapter on CFDP Directives describes available control once the process is running. These options below are given at start up time.

2.1 Socket Setup

The CFDP application uses the ITOS socket library. Refer to documentation on socket_instance_setup (in the ITOS Sockets API Documentaion) for information on configuring the CFDP socket connections.

Socket names are used in configuring the sockets connections. The names must be as follows:

FUNCTION	NAME
CFDP Output Socket Name:	cfdpoutput
CFDP Input Socket Name:	cfdpinput
CFDP STOL Socket Name:	cfdpstol

Options only relevant for TCP client sockets (meaningless for TCP servers):

‘`--host` *<host>*’

‘`--max-retries` *<mr>*’

‘`--retry-interval` *<int>*’

Options only relevant for TCP server sockets (meaningless for TCP clients):

‘`--keep-listening`’

Refer to cfdp_ground.proc and cfdp_spacecraft.proc for examples.

2.2 Command Line Options

‘`-bypasscmd`’

```
Output packets directly to specified socket, bypasses
command subsystem.  If using ITOS as the spacecraft for
testing this option is required.  Default is not to
bypass.
```

‘`-debug`’

```
Turn debug on, default is off.  *** FOR TESTING ONLY ***
Voluminous output will flood the displays with this
option.  The computer may hang.  Use with care.
```

‘`-format`’

```
where input format is one of::
```

‘packet’ telemetry packet

‘frame’ telemetry frame
NOTE: Not implemented yet.

‘rawpdu’ four byte length, data

default is packet

‘-id <i>’

where i is the local CFDP node/entity ID.

‘-nodetomnemonic <x> <y>’

Node ID to mnemonic mapping, x is node ID, y is mnemonic.
Node ID is in CFDP "put" command line, it is node ID of
destination. It is mapped to the mnemonic given here.
That command mnemonic is used to retrieve the application
ID and other formatting information from the database.

‘-packet_second_head <x>’

where x is size of packet secondary header (in bytes).
This must agree with the offset for the DATA field of
all commands that are supported.

‘-socket <socket_name>’

see previous section for discussion of socket setup. Three
sockets are required: input, output, and STOL. The ‘-socket’
option marks the start of a new socket setup. All socket
configuration for a given socket must immediately follow
that switch. Another ‘-socket’ or any other command line
option marks the end of setup for that socket.

‘-test’

resets statistics when a a new "session" is detected. This
is detemined to be on the transition from no active transactions
to having one or more active transactions. Default is off.

3 CFDP Directives

3.1 Use of STOL for CFDP Directives

CFDP directives are provided for real-time control of CFDP operations. See Chapter on Invocation for detailed information on initial configuration. As mentioned previously the STOL write directive is used to send these directives to the CFDP process. A typical entry in the STOL GUI would be as follows:

‘`write (cfdp_app) "cfdp reset"`’

Note: This assumes the global variable <*cfdp_app*> was created in the STOL proc used to start the CFDP application and used to open the STOL-CFDP connection.

All entries are case insensitive but must start with "cfdp".

3.2 CFDP cancel

```
Cancel causes the specified transaction(s) to be canceled.  The
transactions cannot be be restarted.

‘CFDP cancel 23_808’

    cancels the transaction which CFDP node 23 started and
    assigned transaction-sequence-number = 808.  (a transaction
    is uniquely identified by the combination of the Source-ID
    and trans-seq-number)

‘CFDP cancel 101.158_3’

    cancels the transaction with ID=101.158, trans-seq-number=3.

‘CFDP cancel all’

    cancels all transactions.
```

3.3 CFDP freeze

```
Freezes all active transactions.  All timemouts remain in their
current state.  Use ‘CFDP thaw’ to unfreeze.

‘CFDP freeze’
```

3.4 CFDP off

Terminates the CFDP process.

'CFDP off'

3.5 CFDP put

Put tells the CFDP application to transmit a file. Put Requests use this syntax:

'CFDP put [-class1] <source_file> <destination_id> [dest_file]'

Unless -class1 is explicitly specifed as shown above the transaction is a class 2 (acknowledged, hence reliable) transaction.

'CFDP put abc 23 def'

sends 'abc' to CFDP node 23, and calls it 'def'

'CFDP put abc 23'

sends 'abc' to CFDP node 23, and calls it 'abc'

'CFDP put -class1 abc 23'

same as previous, except Service Class 1 (i.e. no retransmissions by CFDP)

'CFDP put abc 101.158'

if the partners Entity-ID exceeds 1 byte, use dotted-decimal format

'CFDP put -class1 abc 23 def'

Uses service class 1 to transfer 'abc' to node 23 and call it 'def'

3.6 CFDP report

Report tells the CFDP application to report on the specified transaction(s).

'CFDP report 23_808'

Reports on the transaction which CFDP node 23 started and assigned transaction-sequence-number = 808. (a transaction is uniquely identified by the combination of the Source-ID and trans-seq-number)

'CFDP report 101.158_3'

Report on transaction with ID=101.158, trans-seq-number=3.

'CFDP report all'

Reports on all transactions.

3.7 CFDP resume

Resume tells the CFDP application to resume a previously suspended transaction. The transaction picks up where it left off.

'CFDP resume 23_808'

Resumes the transaction which CFDP node 23 started and assigned transaction-sequence-number = 808. (a transaction is uniquely identified by the combination of the Source-ID and trans-seq-number)

'CFDP resume 101.158_3'

Resume transaction with ID=101.158, trans-seq-number=3.

'CFDP resume all'

Resumes all transactions.

3.8 CFDP reset

Resets all CFDP statistics.

'CFDP reset'

3.9 CFDP suspend

Suspend tells the CFDP application to temporarily suspend a transaction. The suspend is reversed by a resume directive, which causes the transaction to pick up where it was left off.

'CFDP suspend 23_808'

```
suspends the transaction which CFDP node 23 started and
assigned transaction-sequence-number = 808.  (a transaction
is uniquely identified by the combination of the Source-ID
and trans-seq-number)
```

'CFDP suspend 101.158_3'

```
ID=101.158, trans-seq-number=3.
```

'CFDP suspend all'

```
suspends all transactions.
```

3.10 CFDP thaw

Thaws all previously frozen transactions.

'CFDP thaw'

3.11 CFDP uplink <on/off>

Enable/disable CFDP uplink. Uplink data is not lost (within the physical limitations of system resources).

'CFDP uplink on'

'CFDP uplink off'

4 ITOS Configuration for CFDP

ITOS CFDP configuration items include:

*.dbx file(s) to create necessary database items

ITOS global mnemonics.

*.page file(s) to display CFDP status,

*.proc file(s) to start the CFDP process,

4.1 CFDP Database

Two '`cfdp.dbx`' files are provided that contain the needed command definitions and telemetry mnemonics to support the CFDP application. LRO mission specific commands are in '`$ITOS_HOME/test/dbx/cfdp.dbx`'. The subsystem identifier and telemetry mnemonics are in '`ITOS_HOME/src/inputs/dbx/cfdp.dbx`'.

4.1.1 CFDP Global Variables

The variable GBL_CMD_CHKSM_PKT must be set to indicate the desired checksum when exchanging packets. When using transfer frames the variable to set is GBL_CMD_CHKSM_TF Also refer to Section 1.2 PDUs Out for information on ITOS_TRANSMIT_HEADER.

4.2 CFDP page

The '`cfdp.page`' displays the CFDP statistics.

4.3 CFDP procs

A STOL proc '`cfdp_ground.proc`' enables the command subsystem and starts the CFDP application configured as a ground station. The STOL proc '`cfdp_spacecraft.proc`' configures and runs a CFDP to act as a spacecraft simulator. It shows examples of opening and writing to the STOL socket.